

- 1) Username, Pwd, Confirm Pwd  
2) Restart

Username : nehaj

Password : 31staigamnet

Full name : Nehaj Jaiswal

email : nehajjais5@gmail.com

03/01/23 To create new job create Build

1. + New Item

2. Name : First job neh

3. Freestyle Project

4.

5. Configure - general


Description: First job in Jenkins

6. Source code management - None

Build Steps - Execute Windows batch cmd

echo "Hello Students: %date% : %time%"

7. Build Now

click on  Console output

1. + New Item

2. Name : Automatic build

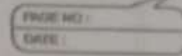
3. <sup>Description:</sup> Build ~~Step~~ Triggers - Build Periodically

\* \* \* \* \*

4. Build Steps : Execute Windows batch cmd  
echo "This is going to trigger every  
minute automatically"

5. Save

C:\Program Files\Git\bin\git.exe



6. wait for 2 min
7. Check Console output.

### Connecting Jenkins with Github

manage Jenkins  
manage Plugins

1. + New Item
2. New Github
3. Free Style
4. ok

5. Description : Connection to git
6. Source Code management - Git
7. Repository URL : Paste URL. (Github)
8. Credentials : [+Add] Jenkins username repository  
Pwd: 31stargaweb  
[Add]
9. Select credentials just chosen
10. Branch Specifier: \*/main
11. Build Steps - Execute using batch cmd  
javaac Hello.java  
java Hello
12. Save.
13. Build Now
14. Click Build

Now make some changes in Hello.java file  
and build again



### Python file

- 1) Do the same as prev
- 2) In Execute window batch command write

`Python -m unittest one.py`

### HTML file

- 1) Do the same as previous files
- 2) In build steps don't choose anything
- 3) Build Now
- 4) click on git-htul
- 5) workspace
- 6) double click hello.html.

- 1) Sign to Google account
- 2) enable 2-step verification in Security.
- 3) Application Prod   
 mail Google account

- 4) manage Jenkins
- 5) Configure system
- 6) Email Notification - SMTP Securer   
 smtp.gmail.com

- 7) Advanced - check use SMTP Auth:   
 Username - google name   
 Password - google pwd generated   
 use SSL

- 8) SMTP Port 465   
 Reply to address - gmail   
 utf-8

- 9) gmail Test configuration   
 Apply & Save

## Role based authorization strategy.

1. ~~Developer~~ → Permissions

1. Manage Jenkins

2. Manage Plugin - Available

2. Installed Plugin -

3. Manage ~~Jenkins~~ Plugin - Available - Search

role based auth and click install without restart

## Configure global security.

1. Authorization - role based strategy

2. apply & save.

use manage Jenkins

3. manage & assign Roles → manage roles

4. Role to add : Developer

Add

QA

Add

DevOps

Add

overall job

5. Give Perm. admin - 1

else 2, read

all - overall read & job read  
job discover & to view configure,  
view create, view read

job discover

job read

view - configure

create read

6. Apply save.

developer - create job,  
delete job,  
build job,  
cancel job, configure job

7. Manage Jenkins → Manage Users

8. Create user

username - qa1  
pwd - qa1

9. user - dev1 ← Pwd Fullname - dev1

2 users under each role. dev, qa, do

10. Dashboard - Manage Jenkins - Manage and

assign - search developer 1 - developer role  
[add]

qa1 - qa role

devops1 - devops role.

11. Apply & Save.



usec...  
developer 1  
developer 2  
qa 1  
qa 2  
deveops 1, deveops 2

two

same part.

PAGE NO:  
DATE:

10/10/23

12. Now open firefox
13. localhost:8080
14. Now open developer 1 & you will not see manage option

- 1) New Item
- 2) Folder → qa team.
- 3) create 2 jobs in that.
- 4) In admin acc - manage Jenkins - manage role
- 5) Item roles  
role to add: QA qa1  
pattern: qa.\*  & role  
create, configure,  
cancel, delete, read job  
Double click on "qa\*!"
- 6) Apply & save
- 7) Assign role.
- 8) type qa1 and
- 9) Apply & save

Jen

## Creating Pipelines in Jenkins (2 ways)

1. using build Pipeline
  2. using pipeline code
- Two syntax are used for defining Jenkinsfile
1. Scripted
  2. Declarative.

### ★ Creating a Jenkins Pipeline using "BuildPipeline" Plugin "

create 3 jobs

1. New Item - Build - FreeStyle  
Test - FreeStyle  
Deploy - FreeStyle.
2. Execute view batch command  
echo "This is to build an Application"
3. Give description
4. Apply & Save.
5. Copy from Build for Test and Deploy jobs.
6. Apply and Save.
7. Manage Plugins  
Search build Pipeline click (Install without restart)
8. Go back to Dashboard.  
Click on Build - Click on Configure.  
Post-build Actions - Build other Projects  
Projects to build: Test
9. Apply and Save.
10. Go to Test in Dashboard. → Configure  
Post-build Actions - Build other Projects  
Projects to build: Deploy.
11. Apply and Save.



Install one more Plugin → Safe Restart.

12. Click on (+)

Name: Pipeline

↳ Build Pipeline View

[Create]

13. Upstream / downstream config

Select - Build.

Apply Save

14. Trigger a Pipeline

→ ★ How to manually trigger a specific job in pipeline.

1. Test → Configure.

Post-build Actions - Build other Project (manual)

Specify Project Name: Deploy

Apply and Save.

★ Now in Pipeline press trigger of Deploy job only then it will build

→ ★ If we need one more job parallel to test

1. + New Item CO (Freestyle)

Description: for checking

Execute with batch cmd

echo "This is to check code"

Dashboard

2. ↳ Build → Configure

Test, CO

Apply and Save

3. Dashboard Pipeline

4. Trigger Pipeline.

13-01-23

PAGE NO.  
DATE:

## Extended mail notification

In extended E-mail notification

1. SMTP Server: smtp.gmail.com

SMTP Port: 465

click Advanced

+ Add User → email id

Password - email generated

Add

Credentials → choose email

use SSL

Default Content Type: HTML

Default Subject:

Default Content:

(tick) Default Triggers: ~~Failure~~ Always (Uncheck failure)

Apply Save

→ Go to FirstJob → Configure

Post-build action: Editable Email Notification

Project from: FirstJob

Project Recipient list: email id.

Project Reply To list: " "

Content type - HTML.

Default Content = ~~Summary~~ \$DEFAULT\_CONTENT

Apply and Save

Go to FirstJob → Build Now

manage



Create a Pipeline using Declarative Syntax.

1. New Item: Declarative Pipeline
2. Choose Pipeline - create
3. Pipeline: Definition: Pipeline script
4. Script (drop down): HelloWorld.
5. Copy and paste in notepad, the code.
6. Apply & save
7. Build Now
8. Configure - change echo in code.  
echo "This is to build an APP"  
echo "This is to test an APP"  
echo "This is to deploy an APP"
9. Give description: pipeline using Declarative syntax
10. Apply & save
11. Build Now.
12. Configure: echo for deploy to check previous

Creating Pipeline by using Jenkinsfile hosted in github.

1. Declarative Pipeline → Configure →
2. copy code in notepad
3. Pipeline: Definition: Pipeline Script from scm
4. SCM: Git
5. Repository URI: Create Jenkinsfile (paste code) after code write Post?

always

emailText body: 'Summary', replyTo: 'emailid',  
Subject: 'Pipe Notify', to: 'email id'

1. Click Pipeline Syntax

echo: Print Message

click generate Pipeline Script

2. Email extended in Sample Std

~~email~~ <sup>TO</sup> id : email id

Subject - Pipeline - Notify

Body - Summary

click generate

[ \* type Jenkins file official website on google

[ \* See Handling failure

3. Apply same - Build Now.

4. Repository URL: Copy URL of repo containing Jenkins file  
give filename as Jenkinsfile

5. Credentials : Email

6. Branches to build \*/main

7. Script Path: Jenkinsfile

8. Apply and Save, Build Now



## Poll SCM

### declarative pipeline

1. Configure: Build Trigger: Poll SCM
2. Schedule: \* \* \* \* \*
3. Apply & Save
4. Go to github - modify any file - commit
5. Refresh Jenkins
- 6.

to on google

### Download Maven:

- 1) Apache Maven - Download - ~~Binary~~ Binary zip archive  
- 3.8.7-bin.zip
- 2) unzip it → Paste in C:\Program Files and Paste
- 3) Go to environment variables.  
New → MAVEN\_HOME (system)  
→ C:\Program Files\apache-maven-3.8.7  
Path - ~~new~~ edit (system)  
↳ New %MAVEN\_HOME%\bin → OK

setting jenkins file

- 4) cmd ~~command~~ mvn -version

- 1) Download Apache Tomcat:

- 2) Tomcat 10 - 64 bit windows zip (download)

- 3) unzip it → go to Program files and paste

- 4) click on tomcat folder - go to bin folder

- 5) double click on startup (batch file)

- 6) go to config → server (double click)

- 7) Connector Port → 8081, C

- 8) config - tomcat - user

```
<role roleName = "manager-script" />
```

```
<user username = "deployer" password = "deployer"
  roleName = "manager-script" />
```

- 9) google localhost:8081

## Creating Hello World.

1) `mvn archetype:generate -DgroupId=com.msrit.app`

→ Automatically triggering Jenkins Jobs using Github Webhook

1) Configure

2) Github hook trigger for GITSCM Polling ☒  
Uncheck all others

3) Apply and save

4) Go to repository → Click on Settings

In left corner click on webhook

↳ click on Add webhook

↳ confirm password.

↳ Payload url : Jenkins url (

↳ Install ngrok (cross platform app) that enables developers to expose a local development server to the internet with minimal effort

↳ Install wind -64 - Download - unzip

open cmd → `ngrok.exe -version`

↳ In cmd → `ngrok.exe http 8080`

↳ copy web interface for forwarding and use it as jenkins url

↳ In Github : Paste this url /github-webhook/

↳ Just the push

↳ Active ☒

↳ add webhook.

Go back to jenkins github - do some modifications in any file of repository

→ Come back to jenkins

→ check if email is there